

determining an identity of input to the programming language editor by the user;

automatically resolving symbolic portions of available ones of a plurality of programming language statements by means of a reverse parse evaluation into a partial program compilation that generates identifiable tokens for each of the at least one segment therein in response to the input being on-demand request by the user;

identifying a present programming language statement and at least one segment of the present programming language statement based on a location of the character position cursor;

determining a finite set of information related to the present programming language statement and at least one segment of the present programming language statement based on the automatically generated partial compilation;

automatically generating an assist window of the finite set of information;

receiving a representation of a selection by the computer programmer from the finite set of information; and

modifying the present programming language statement based at least in part on the selected information;

enabling execution of a editing task in response to the input being a programming language editor command;

enabling a first type of commit of an identified menu item from a selection menu assist window in response to the input being a commit key, wherein the step of enabling a first type of commit includes:

identifying the commit key as a non-delimiter type commit key; and

discarding the commit key;

In re Appln. Of Shulman et al.
Serial No. 08/863, 822

enabling a second type of commit of an identified menu item from a selection menu assist window in response to the input being a commit key, wherein the second type of commit includes:

identifying the commit key as a delimiter type commit key; and

inserting the commit key after the identified menu item in the present programming language statement; and

adding to the present programming language statement at a location of the character position cursor in response to the input being a non-commit key type input character. ---

Please amend **claims 1,12,14,17,25,26,28,33,36-41 and 49** as follows:

Claim 1. (Twice Amended) A computer-readable medium containing computer-executable instructions to perform a method for assisting a computer programmer in real-time to modify a present programming language statement of a computer program, the method comprising:

enabling a programming language editor having a character position cursor and a randomly positionable pointer;

partially compiling available ones of a plurality of programming language statements in said computer program;

defining [generating] a finite set of programming language statement information that is relevant to at least one segment of [modifying] the present programming language statement [;] from among said plurality of programming language statements that is proximate to said character position which allows modification of the programming

In re Appln. Of Shulman et al.
Serial No. 08/863, 822

language statement; and

automatically generating a passive assist window that
contains said finite set of programming language statement
information in a location proximate to said character
position cursor that does not obstruct the current view of
said programming language statement

[receiving a representation of a selection by the
computer programmer from the generated set of programming
language statement information]; and

modifying the present programming language statement
based on at least in part on the selected programming
language statement information; and

automatically removing the passive assist window when
the programming language statement has been amended.

Claim 12. (Twice Amended) A system for passively
assisting a user in real-time to modify a programming
language statement, the system comprising:

a programming language editor having a character
position cursor and a randomly positionable pointer;

means for partially compiling, in an automatic manner,
available ones of a plurality of programming language
statements in the computer program;

means for automatically generating an assist window that
contains a set of programming language statement information
in a location proximate to the character position cursor, the
assist window being selected from [at least one of] a group
[comprised] consisting of[:] a selection menu assist window
and an informational display assist window;

means for automatically receiving a selection by the
user from the set of programming language statement

In re Appln. Of Shulman et al.
Serial No. 08/863, 822

information; and

means for modifying a present programming language statement based at least in part on the selected programming language statement information.

Claim 14. (Twice Amended) The system of claim 12 further comprising a [including:] means for displaying information in an informational display assist window, the information being related to at least one segment of the present programming language statement and selected from a group consisting of [from at least one type of a group comprised of:] a symbol definition, a defined constant, a procedure call map, and an enumerated list.

Claim 17. (Twice Amended) A real-time method for assisting a user to modify a programming language statement in a computer program, the real-time method comprising:

enabling a programming language editor having a character position cursor;

automatically determining an identity of input to the programming language editor by the user;

[continuously] automatically resolving symbolic portions of available ones of a plurality of programming language statements by means of a reverse parse evaluation into a partial program compilation that generates identifiable tokens for each of the at least one segment therein in response to the input being an on-demand request by the user;

identifying a present programming language statement and at least one segment of the present programming language statement based on a location of the character position cursor;

In re Appln. Of Shulman et al.
Serial No. 08/863, 822

determining a finite set of information related to the present programming language statement and [the] at least one segment of the present programming language statement based on the automatically generated partial compilation;

automatically generating an assist window of the finite set of information;

receiving a representation of a selection by the computer programmer from the finite set of information; and

modifying the present programming language statement [to] based at least in part on the selected information;

enabling execution of a editing task in response to the input being a programming language editor command;

enabling a first type of commit of an identified menu item from a selection menu assist window in response to the input being a commit key, wherein the step of enabling a first type of commit includes:

identifying the commit key as a non-delimiter type commit key; and

discarding the commit key;

enabling a second type of commit of an identified menu item from a selection menu assist window in response to the input being a commit key, wherein the second type of commit includes:

identifying the commit key as a delimiter type commit key; and

inserting the commit key after the identified menu item in the present programming language statement; and

adding to the present programming language statement at a location of the character position cursor in response to the input being a non-commit key type input character.

In re Appln. Of Shulman et al.
Serial No. 08/863, 822

Claim 25. (Amended) The system of claim 12, wherein [the] a representation of the user's selection is received in response to the activation of a commit key.

Claim 26. (Amended) The computer-readable medium of claim 52[17], wherein the programming language statement is modified in response to the receipt of an indication that a commit key has been activated.

Claim 28. (Amended) The computer-readable medium of claim 52[17], wherein the representation of the selection is received in response to the activation of a commit key.

Claim 33. (Amended) A method in a computer system for supplementing an incomplete computer programming statement, the method comprising the steps of:

(a) automatically displaying the incomplete computer programming statement;

(b) proximate to the display of [the] an incomplete computer programming statement, displaying a list of one or more textual programmatic entities;

(c) receiving a user input selection of one of the displayed textual programmatic entities; and

(d) adding the selected textual programmatic entity to the displayed statement.

Claim 36. (Amended) The method of claim 33 wherein step (b) displays textual programmatic entities that can validly be included in [the] a partial statement.

Claim 37. (Amended) The method of claim 36[33] wherein the partial statement includes an object identifier, and

In re Appln. Of Shulman et al.
Serial No. 08/863, 822

step (b) displays members of an object identified by an object identifier.

Claim 38. (Amended) The method of claim 36[33] wherein the partial statement includes a portion of a symbol, and wherein step (b) displays complete symbols in which the portion of a symbol is contained.

Claim 39. (Amended) The method of claim 36[33] wherein the partial statement is a partial assignment statement identifying a variable to which a value is to be assigned, and wherein the identified variable has a type, and wherein the type has possible values, and wherein step (b) displays possible values of the type.

Claim 40. (Amended) The method of claim 36[33] wherein the partial statement is a partial function call statement identifying a function, and wherein the identified function has one or more parameters, and wherein step (b) displays parameters of the identified function.

Claim 41. (Amended) A computer-readable medium whose contents cause a computer system to supplement a computer programming statement by performing the steps of:

(a) automatically displaying the computer programming statement;

(b) proximate to the display of [the] an incomplete computer programming statement, displaying one or more textual programmatic entities;

(c) receiving a user input selection of one of the displayed textual programmatic entities; and